CLAIMS

What is claimed:

1. A method for reducing context memory requirements in a multi-tasking system, comprising:

providing a hardware engine in a computer processor,

applying a compression algorithm in said hardware engine to each instance in a multi-instance software system to reduce context memory in said software system.

2. The method of claim 1, wherein said applying comprises applying a generic, lossless compression algorithm that performs an adaptive packing operation.

3. The method of claim 1, wherein said applying comprises:

dividing data in instances of said multi-instance system into blocks; and

for each said instance:

assigning a packing width to a block having a maximum number of significant bits;

encoding, with said compression algorithm, least significant bits of each word in said block into a packed block of said packing width multiplied by a total number of words in said block; and

providing a prefix header at the beginning of each packed block to represent a change in said packing width from said packed block from a packing width of a previous packed block.

4. The method of claim 3, wherein said dividing comprises dividing blocks containing the same number of words.

5. The method of claim 3, wherein said providing said prefix header comprises encoding said prefix as a variable length sequence that uses between one and seven bits.

6. The method of claim 1, wherein said applying comprises encoding each word in a packed block using a lossless compression hardware engine integrated into said processor.

7. The method of claim 3, wherein said encoding comprises performing an adaptive packing operation on said least significant bits.

8. The method of claim 3, further comprising:

expanding said compressed data with a decoder on said hardware engine; and

moving said expanded data from a shared memory on said processor to a local memory on said processor;

processing said data in said channel in accordance with the application running on said processor; and

moving said compressed data from said local memory into said shared memory.

9. The method of claim 3, further comprising:

providing a last block prefix header to a final block of said data, wherein said last block prefix header comprises a last block marker of six bits followed by two bits that define the number of said words contained in the final block.

10. A method for reducing context memory requirements in a multi-tasking system, comprising:

providing a hardware engine in a computer processor,

dividing data in a task of said multi-tasking system into blocks of words;

applying a compression algorithm in said hardware engine to each word to create packed blocks of said words; and

providing a prefix header at the beginning of each packed block to represent a change in packing width from said packed block from a packing width of a previous packed block.

11. The method of claim 10, wherein each block contains the same number of said words.

12. The method of claim 10, further comprising for each said task:

determining a word in a block having a maximum number of significant bits;

assigning a packing width to said block of said maximum number of significant bits;

encoding, with said compression algorithm, least significant bits of each word in said block into a packed block of said packing width multiplied by a total number of words in said block.

13. The method of claim 10, wherein said compression algorithm is lossless compression algorithm.

14. The method of claim 10, further comprising:

expanding said compressed data with a decoder on said hardware engine; and

moving said expanded data from a shared memory on said processor to a local memory on said processor;

processing said data in said channel in accordance with the application running on said processor; and

moving said compressed data from said local memory into said shared memory.

15. The method of claim10, further comprising:

providing a last block prefix header to a final block of said data, wherein said last block prefix header comprises a last block marker of six bits followed by two bits that define the number of said words contained in the final block.